### Division of Labour

A group of four developers were working together on this first project. The group met several times in the run up to the client interface submission. In the initial meeting we discussed suitable choices for our group, from the given possibilities, based on the group members' previous experiences. We decided on the Synchronous Chat Service as this played to several group members' strengths while also offering new experience for others.

During the next meeting we decided on the main tasks, the project plan for when these tasks would need to be submitted, and the division of labour. The interface was broken down into XHTML tasks, scripting, and the Java applet we decided to use for the IRC Chat interface. To try and utilise more XHTML, we decided to include a cover page and guide page. These tasks were shared between Dean, covering most of the CSS, and Ximeng, working more on XHTML. The client interface would also include a login form to post data to the applet which would also need Javascript to allow for form validation and the use of optional fields. This form was handled by David. Simon was elected to handle most of the Java applet development as he had the most experience at producing a similar service in the past.

Moving forward, the group now had most of the pages and content for the service developed, with more of the back-end in terms of scripting and the applet still needed developed. Upon receipt of feedback from the course supervisor, the group discussed how to proceed while utilising this new information. It was stated that more XHTML was needed, especially around the client interface, so the group decided to see where we could add more to try and include as much as possible. Dean was then tasked with writing the PHP script to pass the POST data from the login form to the client interface, while Simon continued to touch up the applet itself.

Getting ready for the final submission, each group member then chose a section to write for the final report, dividing the four sections of the main report between us while all contributing to the testing documentation as well.

Each member put in plenty of effort to contribute to the group. A git repository was well used, allowing each member to inform their fellows of changes and keep track of everyone's progress. Each group member was communicative and completed their assigned tasks properly.

## How the system works

The system uses a HTML form to gain the login details for the IRC server, including Javascript validation along with a PHP backend to parse the parameters in the way that the client needs them (such as adding optional parameters with default values, or choosing common values from a list). The PHP script then builds a page containing a Java Applet which hosts the IRC client and passing in the required parameters.

The IRC applet itself is digitally signed to allow it to connect to other servers, and uses a socket to connect to the IRC server. A BufferedReader and a PrintWriter write and read from the socket, and separate threads handle the background work involved with these. Data from IRC is parsed, and then passed to the interface for display. Input from the client is taken, checked to see if it is prefixed with a / (to indicate a raw command), and forwarded to the IRC server if necessary. An “active channel” variable indicates the channel which the user is currently talking in, and can be modified with the /channel client-specific command.